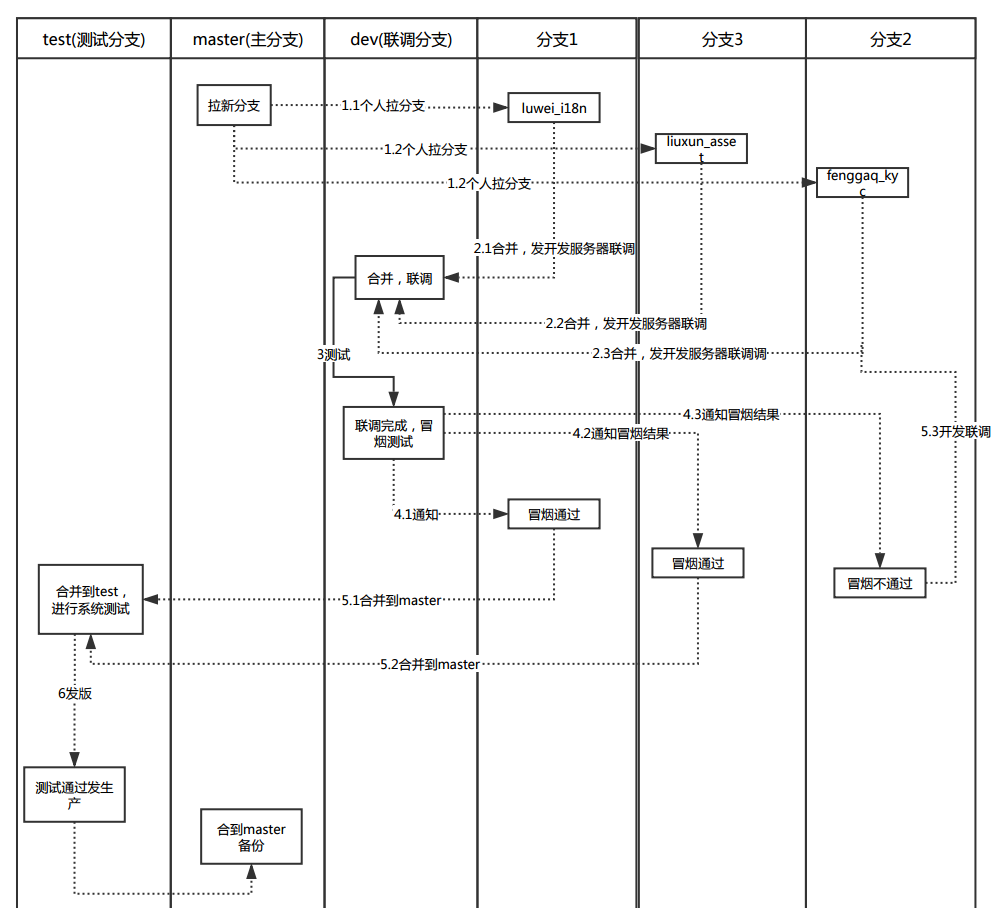
# Git版本控制规范V1.1

## 分支管理规范说明

1. 主分支master、开发分支dev。
2. 拉新分支命名规范：分支创建人姓名拼音+下划线+功能点简称，如：luwei\_i18n，含义是：luwei创建国际化分支。
3. 项目稳定阶段，可从master拉取新分支，大家在一个分支上开发，方便与web、app联调各接口，联调完成之后，合并到主分支master上进行测试。
4. 项目不稳定、变更频繁，需求较多，则按以下流程控制：
5. 业务模块开发人员从master上拉取分支，在自己的分支上进行开发。
6. 需要发到开发服务器与web、app联调的，则将新拉取的分支合并到开发分支dev上。
7. 将dev分支打包发到开发服务器，供其它小组联调。
8. 功能完成之后，测试人员进行冒烟测试。
9. 冒烟通过，则将新分支合并到test分支，将test分支打包发到测试服务器进行系统测试。
10. Test分支测试通过后，上线发布，发布完成合并到master备份

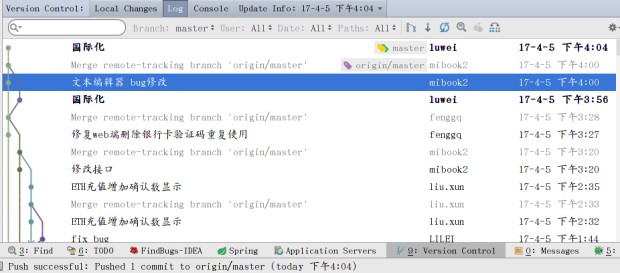
过程如下图所示：



1. 分支使用完毕，上线完成之后，需清理分支，即在确认不使用后，删除该分支，避免分支过多，产生误导。
2. 若小组成员新做需求有相互影响，相互依赖的，可在一个分支上进行开发。

## 代码提交规范

1. 代码提交前，先pull，避免出现冲突
2. 代码提交时，描述清楚提交的内容，有修复bug的，需把bug号填写进去；新功能的需描述实现\*\*功能，或实现\*\*接口。
3. 提交人必须是开发人员自己的姓名，便于查找日志，查看模块、bug负责人，禁止出现无法识别的用户，如：mibook2，具体示例如下图所示：



## Tag标签管理说明

1. 所有上线的代码都在主分支(master)上，**每上完一个版本后，打一个tag，名称命名为本次上线的版本号，说明填写本次上线内容的概述**。
2. 新需求改动较大时，打一个tag，名称命名为V版本号\_备份，说明填写\*\*需求改动备份。

示例如下：

* V1.1.3 首页改版、资讯；
* V1.1.4 支付宝、审核权限；
* V1.1.5 帮助中心、权限、M站；

效果如下图所示：

